Dependency Theory

COMPSCI 2DB3: Databases

Jelle Hellings

Department of Computing and Software
McMaster University

Winter 2023

Recap

- The Entity-Relationship Model.
  High-level modeling of data.
- SQL: The Structured Qery Language.
  Querying relational data in practice.
- The Relational Data Model and SQL.
  Creating relational tables from high-level models.
- The Relational Algebra.
  Abstract easy-to-manipulate querying of relational data.

Outlook

- Dependency Theory.
- Decomposition and Normal Forms.
- Concurrency Control.

Outlook

- Dependency Theory.
- Decomposition and Normal Forms.
- Concurrency Control.

Topics of Interest

Next step: Formalizing constraints

We have see primary keys, foreign keys, checks, ...

Next step: Formalizing constraints

We have seen primary keys, foreign keys, checks, ...

How can we reason about (typical) constraints?

Warning: Proofs incoming

Remember Discrete Mathematics!

Warning: Proofs incoming

Remember Discrete Mathematics!

I will try to keep things simple.

Warning: Proofs incoming

Remember Discrete Mathematics!

I will try to keep things simple.

If I go too fast—please press the brake.
E.g., weird notation, steps I overlooked, …

Let us start with an example

student(<u>sid</u>, name, age, birthdate, program, department)

Let us start with an example

student(<u>sid</u>, name, age, birthdate, program, department)

Question: Rate my table!

Vote at https://strawpoll.com/dycss6a57.
Or: go to https://strawpoll.live and use the code **277712**.

Let us start with an example

student(<u>sid</u>, name, age, birthdate, program, department)

- Attribute sid is the primary key: sid determines all attributes.
  Students with the same sid are the same student.

Let us start with an example

student(sid, name, age, birthdate, program, department)

- Attribute sid is the primary key: sid determines all attributes.
  Students with the same sid are the same student.
- Attribute birthdate determines age.
  Students with the same birthdate have the same age.

Let us start with an example

student(sid, name, age, birthdate, program, department)

- Attribute sid is the primary key: sid determines all attributes.
  Students with the same sid are the same student.
- Attribute birthdate determines age.
  Students with the same birthdate have the same age.
- Each program is organized by a department.
  Students in the same program belong to the same department.

Functional dependency over relation schema $\mathbf{R}$

$$X \longrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{).}$$

Functional dependency over relation schema **R**

$$X \longrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{).}$$

Informal

"Attributes X determine Y":
If two rows in an instance of **R** have the same values for attributes X,
Then these rows have the same values for attributes Y.

Functional dependency over relation schema **R**

$$X \longrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{).}$$

Informal

"Attributes X determine Y":
If two rows in an instance of **R** have the same values for attributes X,
Then these rows have the same values for attributes Y.

Formal

Let $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_m\}$.
For every instance $\mathcal{I}$ of **R** and every pair of rows $r_1, r_2 \in \mathcal{I}$, we have:

$$(r_1[x_1] = r_2[x_1] \wedge \cdots \wedge r_1[x_n] = r_2[x_n]) \implies (r_1[y_1] = r_2[y_1] \wedge \cdots \wedge r_1[y_m] = r_2[y_m]).$$

Functional dependency over relation schema **R**

$$X \longrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{).}$$

Informal

"Attributes X determine Y":
If two rows in an instance of **R** have the same values for attributes X,
Then these rows have the same values for attributes Y.

Formal

Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$.
For every instance $\mathcal{I}$ of **R** and every pair of rows $r_1, r_2 \in \mathcal{I}$, we have:

$$\underbrace{(r_1[x_1] = r_2[x_1] \land \cdots \land r_1[x_n] = r_2[x_n])}_{\text{"same values for attributes } X\text{"}} \implies (r_1[y_1] = r_2[y_1] \land \cdots \land r_1[y_m] = r_2[y_m])\,.$$

Functional dependency over relation schema **R**

$$X \longrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{).}$$

Informal

"Attributes X determine Y":
If two rows in an instance of **R** have the same values for attributes X,
Then these rows have the same values for attributes Y.

Formal

Let $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_m\}$.
For every instance $\mathcal{I}$ of **R** and every pair of rows $r_1, r_2 \in \mathcal{I}$, we have:

$$\underbrace{(r_1[x_1] = r_2[x_1] \land \cdots \land r_1[x_n] = r_2[x_n])}_{\text{“same values for attributes } X\text{”}} \implies \underbrace{(r_1[y_1] = r_2[y_1] \land \cdots \land r_1[y_m] = r_2[y_m])}_{\text{“same values for attributes } Y\text{”}}.$$

Functional dependency over relation schema **R**

$$X \longrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{).}$$

Informal

"Attributes X determine Y":
If two rows in an instance of **R** have the same values for attributes X,
Then these rows have the same values for attributes Y.

Formal

Let $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_m\}$.
For every instance $\mathcal{I}$ of **R** and every pair of rows $r_1, r_2 \in \mathcal{I}$, we have:

$$\underbrace{(r_1[X] = r_2[X])}_{\text{"same values for attributes } X\text{"}} \quad \Longrightarrow \quad \underbrace{(r_1[Y] = r_2[Y])}_{\text{"same values for attributes } Y\text{"}}.$$

Let us continue with our example

student(sid, name, age, birthdate, program, department)

Let us continue with our example

student(sid, name, age, birthdate, program, department)

- Attribute sid is the primary key: sid determines all attributes.
  "sid $\longrightarrow$ name, age, birthdate, program, department".

Let us continue with our example

student(sid, name, age, birthdate, program, department)

- Attribute sid is the primary key: sid determines all attributes.
  "sid $\longrightarrow$ name, age, birthdate, program, department".
- Attribute birthdate determines age.
  "birthdate $\longrightarrow$ age".

Let us continue with our example

student(sid, name, age, birthdate, program, department)

- Attribute sid is the primary key: sid determines all attributes.
  "sid $\longrightarrow$ name, age, birthdate, program, department".
- Attribute birthdate determines age.
  "birthdate $\longrightarrow$ age".
- Each program is organized by a department.
  "program $\longrightarrow$ department".

Let us reason with our example

student(<u>sid</u>, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

Vote at `https://strawpoll.com/1284xkha3`.
Or: go to `https://strawpoll.live` and use the code **535106**.

Let us reason with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

Let us reason with our example

student(<u>sid</u>, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

By definition: we have $BP \longrightarrow AD$ if we have $r_1[BP] = r_2[BP] \implies r_1[AD] = r_2[AD]$ for every instance $\mathcal{I}$ of **student** and every pair of rows $r_1, r_2 \in \mathcal{I}$.

Let us reason with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

By definition: we have $BP \longrightarrow AD$ if we have $r_1[BP] = r_2[BP] \implies r_1[AD] = r_2[AD]$ for every instance $\mathcal{I}$ of **student** and every pair of rows $r_1, r_2 \in \mathcal{I}$.

Let us reason with our example

student(<u>sid</u>, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of **student** such that $r_1[BP] = r_2[BP]$.

Let us reason with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of **student** such that $r_1[BP] = r_2[BP]$.

(proof details)

Hence, $r_1[AD] = r_2[AD]$ holds.

Let us reason with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of **student** such that $r_1[BP] = r_2[BP]$.

By $r_1[BP] = r_2[BP]$, we have $r_1[B] = r_2[B]$ and $r_1[P] = r_2[P]$.

Hence, $r_1[AD] = r_2[AD]$ holds.

Let us reason with our example

student(<u>sid</u>, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of **student** such that $r_1[BP] = r_2[BP]$.

By $r_1[BP] = r_2[BP]$, we have $r_1[B] = r_2[B]$ and $r_1[P] = r_2[P]$.
Using $B \longrightarrow A$ and $r_1[B] = r_2[B]$, we conclude $r_1[A] = r_2[A]$.

Hence, $r_1[AD] = r_2[AD]$ holds.

Let us reason with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of **student** such that $r_1[BP] = r_2[BP]$.

By $r_1[BP] = r_2[BP]$, we have $r_1[B] = r_2[B]$ and $r_1[P] = r_2[P]$.
Using $B \longrightarrow A$ and $r_1[B] = r_2[B]$, we conclude $r_1[A] = r_2[A]$.
Using $P \longrightarrow D$ and $r_1[P] = r_2[P]$, we conclude $r_1[D] = r_2[D]$.

Hence, $r_1[AD] = r_2[AD]$ holds.

Let us reason with our example

student(<u>sid</u>, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations B (birthdate), P (program), A (age), and D (department).

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of **student** such that $r_1[BP] = r_2[BP]$.

By $r_1[BP] = r_2[BP]$, we have $r_1[B] = r_2[B]$ and $r_1[P] = r_2[P]$.
Using $B \longrightarrow A$ and $r_1[B] = r_2[B]$, we conclude $r_1[A] = r_2[A]$.
Using $P \longrightarrow D$ and $r_1[P] = r_2[P]$, we conclude $r_1[D] = r_2[D]$.
By $r_1[A] = r_2[A]$ and $r_1[D] = r_2[D]$, we have $r_1[AD] = r_2[AD]$.

Hence, $r_1[AD] = r_2[AD]$ holds.

Implication of dependencies

Definition

Let $\mathfrak{S}$ be a set of dependencies and D be a dependency over relation schema $\mathbf{R}$.

We say that $\mathfrak{S}$ *implies* D if, for every instance $\mathcal{I}$ of $\mathbf{R}$ we have,

if $\mathcal{I}$ satisfies $\mathfrak{S}$, then it also satisfies D.

Implication of dependencies

Definition

Let $\mathfrak{S}$ be a set of dependencies and D be a dependency over relation schema $\mathbf{R}$.

We say that $\mathfrak{S}$ *implies* D if, for every instance $\mathcal{I}$ of $\mathbf{R}$ we have,

$$\text{if } \mathcal{I} \text{ satisfies } \mathfrak{S}, \text{ then it also satisfies } D.$$

Informally: $\mathfrak{S}$ implies D if we can "prove" that D holds just from $\mathfrak{S}$.

Implication of dependencies

Definition

Let $\mathfrak{S}$ be a set of dependencies and D be a dependency over relation schema $\mathbf{R}$.

We say that $\mathfrak{S}$ *implies* D if, for every instance $\mathcal{I}$ of $\mathbf{R}$ we have,

if $\mathcal{I}$ satisfies $\mathfrak{S}$, then it also satisfies D.

Informally: $\mathfrak{S}$ implies D if we can "prove" that D holds just from $\mathfrak{S}$.

If $\mathfrak{S}$ implies D, then we write $\mathfrak{S} \models D$.

Implication of dependencies

Definition

Let $\mathfrak{S}$ be a set of dependencies and D be a dependency over relation schema $\mathbf{R}$.

We say that $\mathfrak{S}$ *implies* D if, for every instance $\mathcal{I}$ of $\mathbf{R}$ we have,

if $\mathcal{I}$ satisfies $\mathfrak{S}$, then it also satisfies D.

Informally: $\mathfrak{S}$ implies D if we can "prove" that D holds just from $\mathfrak{S}$.

If $\mathfrak{S}$ implies D, then we write $\mathfrak{S} \models D$.

Example

- $\mathfrak{S} = \{$"birthdate $\longrightarrow$ age", "program $\longrightarrow$ department"$\}$.
- $D =$ "birthdate, program $\longrightarrow$ age, department".

We have $\mathfrak{S} \models D$ (proven on previous slide).

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps.

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps & prove correctness of these rules.

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps & prove correctness of these rules.

Example: The Union rule

Let X, Y, Z be sets of attributes of relation schema $\mathbf{R}$. We have

$$\text{if } X \longrightarrow Y \text{ and } X \longrightarrow Z\text{, then } X \longrightarrow YZ.$$

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps & prove correctness of these rules.

Example: The Union rule

Let X, Y, Z be sets of attributes of relation schema $\mathbf{R}$. We have

$$\text{if } X \longrightarrow Y \text{ and } X \longrightarrow Z\text{, then } X \longrightarrow YZ.$$

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps & prove correctness of these rules.

Example: The Union rule

Let X, Y, Z be sets of attributes of relation schema $\mathbf{R}$. We have

$$\text{if } X \longrightarrow Y \text{ and } X \longrightarrow Z, \text{ then } X \longrightarrow YZ.$$

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

By definition: we have $X \longrightarrow YZ$ if we have $r_1[X] = r_2[X] \implies r_1[YZ] = r_2[YZ]$ for every instance $\mathcal{I}$ of $\mathbf{R}$ and every pair of rows $r_1, r_2 \in \mathcal{I}$.

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps & prove correctness of these rules.

Example: The Union rule

Let X, Y, Z be sets of attributes of relation schema $\mathbf{R}$. We have

$$\text{if } X \longrightarrow Y \text{ and } X \longrightarrow Z, \text{ then } X \longrightarrow YZ.$$

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of $\mathbf{R}$ such that $r_1[X] = r_2[X]$.

(proof details)

Hence, $r_1[YZ] = r_2[YZ]$ holds.

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps & prove correctness of these rules.

Example: The Union rule

Let X, Y, Z be sets of attributes of relation schema $\mathbf{R}$. We have

$$\text{if } X \longrightarrow Y \text{ and } X \longrightarrow Z \text{, then } X \longrightarrow YZ.$$

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of $\mathbf{R}$ such that $r_1[X] = r_2[X]$.

Using $X \longrightarrow Y$ and $r_1[X] = r_2[X]$, we conclude $r_1[Y] = r_2[Y]$.

Hence, $r_1[YZ] = r_2[YZ]$ holds.

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps & prove correctness of these rules.

Example: The Union rule

Let X, Y, Z be sets of attributes of relation schema **R**. We have

$$\text{if } X \longrightarrow Y \text{ and } X \longrightarrow Z\text{, then } X \longrightarrow YZ.$$

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of **R** such that $r_1[X] = r_2[X]$.

Using $X \longrightarrow Y$ and $r_1[X] = r_2[X]$, we conclude $r_1[Y] = r_2[Y]$.
Using $X \longrightarrow Z$ and $r_1[X] = r_2[X]$, we conclude $r_1[Z] = r_2[Z]$.

Hence, $r_1[YZ] = r_2[YZ]$ holds.

Simplifying proofs: Using inference rules

Idea: Make rules that cover common prove steps & prove correctness of these rules.

Example: The Union rule

Let X, Y, Z be sets of attributes of relation schema $\mathbf{R}$. We have

$$\text{if } X \longrightarrow Y \text{ and } X \longrightarrow Z\text{, then } X \longrightarrow YZ.$$

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of $\mathbf{R}$ such that $r_1[X] = r_2[X]$.

Using $X \longrightarrow Y$ and $r_1[X] = r_2[X]$, we conclude $r_1[Y] = r_2[Y]$.
Using $X \longrightarrow Z$ and $r_1[X] = r_2[X]$, we conclude $r_1[Z] = r_2[Z]$.
By $r_1[Y] = r_2[Y]$ and $r_1[Z] = r_2[Z]$, we have $r_1[YZ] = r_2[YZ]$.

Hence, $r_1[YZ] = r_2[YZ]$ holds.

Using an inference rules: the Union rule

student(sid, name, age, birthdate, program, department)

Prove $\{$"sid $\longrightarrow$ name", "sid $\longrightarrow$ age"$\} \models$ "sid $\longrightarrow$ name, age".

Using an inference rules: the Union rule

student(sid, name, age, birthdate, program, department)

Prove {"sid $\longrightarrow$ name", "sid $\longrightarrow$ age"} $\models$ "sid $\longrightarrow$ name, age".

The Union rule: if $X \longrightarrow Y$ and $X \longrightarrow Z$, then $X \longrightarrow YZ$.

Using an inference rules: the Union rule

student(sid, name, age, birthdate, program, department)

Prove {"sid $\longrightarrow$ name", "sid $\longrightarrow$ age"} $\models$ "sid $\longrightarrow$ name, age".

The Union rule: if $X \longrightarrow Y$ and $X \longrightarrow Z$, then $X \longrightarrow YZ$.

Apply the Union rule with $X = \{\text{sid}\}$, $Y = \{\text{name}\}$, and $Z = \{\text{age}\}$.

Which interference rules do we need?

We can make as many as we want.

Which interference rules do we need?

We can make as many as we want.

- The Union-3 rule: if $X \longrightarrow Y_1$ and $X \longrightarrow Y_2$ and $X \longrightarrow Y_3$, then $X \longrightarrow Y_1 Y_2 Y_3$.
- The Union-4 rule: if $X \longrightarrow Y_1, \ldots, X \longrightarrow Y_4$, then $X \longrightarrow Y_1 Y_2 Y_3 Y_4$.
- The Union-5 rule: if $X \longrightarrow Y_1, \ldots, X \longrightarrow Y_5$, then $X \longrightarrow Y_1 Y_2 Y_3 Y_4 Y_5$.
- The Union-6 rule: if $X \longrightarrow Y_1, \ldots, X \longrightarrow Y_6$, then $X \longrightarrow Y_1 Y_2 Y_3 Y_4 Y_5 Y_6$.

  $\vdots$

- The Union-i rule: if $X \longrightarrow Y_1, \ldots, X \longrightarrow Y_i$, then $X \longrightarrow Y_1 Y_2 \ldots Y_i$.

Which interference rules do we need?

We can make as many as we want.

- The Union-3 rule: if $X \longrightarrow Y_1$ and $X \longrightarrow Y_2$ and $X \longrightarrow Y_3$, then $X \longrightarrow Y_1 Y_2 Y_3$.
- The Union-4 rule: if $X \longrightarrow Y_1, \ldots, X \longrightarrow Y_4$, then $X \longrightarrow Y_1 Y_2 Y_3 Y_4$.
- The Union-5 rule: if $X \longrightarrow Y_1, \ldots, X \longrightarrow Y_5$, then $X \longrightarrow Y_1 Y_2 Y_3 Y_4 Y_5$.
- The Union-6 rule: if $X \longrightarrow Y_1, \ldots, X \longrightarrow Y_6$, then $X \longrightarrow Y_1 Y_2 Y_3 Y_4 Y_5 Y_6$.

  $\vdots$

- The Union-i rule: if $X \longrightarrow Y_1, \ldots, X \longrightarrow Y_i$, then $X \longrightarrow Y_1 Y_2 \ldots Y_i$.

These rules are *pointless*: We can do the same with the Union rule!

Criteria for a good set of inference rules

Criteria for a good set of inference rules

Sound Every rule must be correct.

Criteria for a good set of inference rules

Sound Every rule must be correct.
Can we prove the correctness of the rule?

Criteria for a good set of inference rules

Sound Every rule must be correct.
Can we prove the correctness of the rule?

Complete We must be able to derive everything that holds.

Criteria for a good set of inference rules

Sound Every rule must be correct.
Can we prove the correctness of the rule?

Complete We must be able to derive everything that holds.
Can we derive D from $\mathfrak{S}$ using the rules whenever $\mathfrak{S} \models D$ holds?

Criteria for a good set of inference rules

Sound Every rule must be correct.
Can we prove the correctness of the rule?

Complete We must be able to derive everything that holds.
Can we derive D from $\mathfrak{S}$ using the rules whenever $\mathfrak{S} \models D$ holds?

Independent We need every rule for some derivations.

Criteria for a good set of inference rules

Sound Every rule must be correct.
Can we prove the correctness of the rule?

Complete We must be able to derive everything that holds.
Can we derive D from $\mathfrak{S}$ using the rules whenever $\mathfrak{S} \models D$ holds?

Independent We need every rule for some derivations.
Are there any facts that hold we cannot derive after removing a rule?

Armstrong's Axioms

A set of three good interference rules for functional dependencies.

Armstrong's Axioms

A set of three good interference rules for functional dependencies.

Reflexivity If $Y \subseteq X$, then $X \longrightarrow Y$.

Armstrong's Axioms

A set of three good interference rules for functional dependencies.

Reflexivity If $Y \subseteq X$, then $X \longrightarrow Y$.

Augmentation If $X \longrightarrow Y$ then $XZ \longrightarrow YZ$ for any Z.

Armstrong's Axioms

A set of three good interference rules for functional dependencies.

Reflexivity	If $Y \subseteq X$, then $X \longrightarrow Y$.
Augmentation	If $X \longrightarrow Y$ then $XZ \longrightarrow YZ$ for any Z.
Transitivity	If $X \longrightarrow Y$ and $Y \longrightarrow Z$, then $X \longrightarrow Z$.

Armstrong's Axioms

A set of three good interference rules for functional dependencies.

Reflexivity	If $Y \subseteq X$, then $X \longrightarrow Y$.
Augmentation	If $X \longrightarrow Y$ then $XZ \longrightarrow YZ$ for any Z.
Transitivity	If $X \longrightarrow Y$ and $Y \longrightarrow Z$, then $X \longrightarrow Z$.

These inference rules are sound, complete, and independent.

Let us use Armstrong's Axioms with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Let us use Armstrong's Axioms with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Let us use Armstrong's Axioms with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Prove $\{B \longrightarrow A, P \longrightarrow D\} \models BP \longrightarrow AD$.

Let us use Armstrong's Axioms with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Prove $\{B \longrightarrow A, P \longrightarrow D\} \models BP \longrightarrow AD$.

Assume $B \longrightarrow A$ and $P \longrightarrow D$.

(proof details)

Hence, $BP \longrightarrow AD$.

Let us use Armstrong's Axioms with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Prove $\{B \longrightarrow A, P \longrightarrow D\} \models BP \longrightarrow AD$.

Assume $B \longrightarrow A$ and $P \longrightarrow D$.

Apply *Augmentation* on $B \longrightarrow A$ with P to derive $BP \longrightarrow AP$.

Hence, $BP \longrightarrow AD$.

Let us use Armstrong's Axioms with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Prove $\{B \longrightarrow A, P \longrightarrow D\} \models BP \longrightarrow AD$.

Assume $B \longrightarrow A$ and $P \longrightarrow D$.

Apply *Augmentation* on $B \longrightarrow A$ with P to derive $BP \longrightarrow AP$.
Apply *Augmentation* on $P \longrightarrow D$ with A to derive $AP \longrightarrow AD$.

Hence, $BP \longrightarrow AD$.

Let us use Armstrong's Axioms with our example

student(sid, name, age, birthdate, program, department)

Question: Does "birthdate, program $\longrightarrow$ age, department" hold?

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Prove $\{B \longrightarrow A, P \longrightarrow D\} \models BP \longrightarrow AD$.

Assume $B \longrightarrow A$ and $P \longrightarrow D$.

Apply *Augmentation* on $B \longrightarrow A$ with P to derive $BP \longrightarrow AP$.
Apply *Augmentation* on $P \longrightarrow D$ with A to derive $AP \longrightarrow AD$.
Apply *Transitivity* on $BP \longrightarrow AP$ and $AP \longrightarrow AD$ to derive $BP \longrightarrow AD$.

Hence, $BP \longrightarrow AD$.

Adding Union to Armstrong's Axioms

We will show that the Union rule is sound, but *not* independent.

Adding Union to Armstrong's Axioms

We will show that the Union rule is sound, but *not* independent.

How: Prove the Union rule via Armstrong's Axioms

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ.$

Adding Union to Armstrong's Axioms

We will show that the Union rule is sound, but *not* independent.

How: Prove the Union rule via Armstrong's Axioms

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Adding Union to Armstrong's Axioms

We will show that the Union rule is sound, but *not* independent.

How: Prove the Union rule via Armstrong's Axioms

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Assume $X \longrightarrow Y$ and $X \longrightarrow Z$.

(proof details)

Hence, $X \longrightarrow YZ$.

Adding Union to Armstrong's Axioms

We will show that the Union rule is sound, but *not* independent.

How: Prove the Union rule via Armstrong's Axioms

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Assume $X \longrightarrow Y$ and $X \longrightarrow Z$.

Apply *Augmentation* on $X \longrightarrow Y$ with X to derive $X \longrightarrow XY$.

Hence, $X \longrightarrow YZ$.

Adding Union to Armstrong's Axioms

We will show that the Union rule is sound, but *not* independent.

How: Prove the Union rule via Armstrong's Axioms

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Assume $X \longrightarrow Y$ and $X \longrightarrow Z$.

Apply *Augmentation* on $X \longrightarrow Y$ with X to derive $X \longrightarrow XY$.
Apply *Augmentation* on $X \longrightarrow Z$ with Y to derive $XY \longrightarrow YZ$.

Hence, $X \longrightarrow YZ$.

Adding Union to Armstrong's Axioms

We will show that the Union rule is sound, but *not* independent.

How: Prove the Union rule via Armstrong's Axioms

To prove: $\{X \longrightarrow Y, X \longrightarrow Z\} \models X \longrightarrow YZ$.

Assume $X \longrightarrow Y$ and $X \longrightarrow Z$.

Apply *Augmentation* on $X \longrightarrow Y$ with X to derive $X \longrightarrow XY$.
Apply *Augmentation* on $X \longrightarrow Z$ with Y to derive $XY \longrightarrow YZ$.
Apply *Transitivity* on $X \longrightarrow XY$ and $XY \longrightarrow YZ$ to derive $X \longrightarrow YZ$.

Hence, $X \longrightarrow YZ$.

The Decomposition rule

$$\text{if } X \longrightarrow YZ, \text{ then } X \longrightarrow Y \text{ and } X \longrightarrow Z.$$

We will show that the Decomposition rule is sound, but *not* independent.

The Decomposition rule

$$\text{if } X \longrightarrow YZ, \text{ then } X \longrightarrow Y \text{ and } X \longrightarrow Z.$$

We will show that the Decomposition rule is sound, but *not* independent.

How: Prove the Decomposition rule via Armstrong's Axioms

To prove: $\{X \longrightarrow YZ\} \models X \longrightarrow Y$ and $\{X \longrightarrow YZ\} \models X \longrightarrow Z$.

The Decomposition rule

$$\text{if } X \longrightarrow YZ \text{, then } X \longrightarrow Y \text{ and } X \longrightarrow Z.$$

We will show that the Decomposition rule is sound, but *not* independent.

How: Prove the Decomposition rule via Armstrong's Axioms

To prove: $\{X \longrightarrow YZ\} \models X \longrightarrow Y$ and $\{X \longrightarrow YZ\} \models X \longrightarrow Z$.

Assume $X \longrightarrow YZ$.

(proof details)

Hence, $X \longrightarrow Y$

The Decomposition rule

$$\text{if } X \longrightarrow YZ, \text{ then } X \longrightarrow Y \text{ and } X \longrightarrow Z.$$

We will show that the Decomposition rule is sound, but *not* independent.

How: Prove the Decomposition rule via Armstrong's Axioms

To prove: $\{X \longrightarrow YZ\} \models X \longrightarrow Y$ and $\{X \longrightarrow YZ\} \models X \longrightarrow Z$.

Assume $X \longrightarrow YZ$.

Apply *Reflexivity* on $Y \subseteq YZ$ to derive $YZ \longrightarrow Y$.

Hence, $X \longrightarrow Y$

The Decomposition rule

$$\text{if } X \longrightarrow YZ, \text{ then } X \longrightarrow Y \text{ and } X \longrightarrow Z.$$

We will show that the Decomposition rule is sound, but *not* independent.

How: Prove the Decomposition rule via Armstrong's Axioms

To prove: $\{X \longrightarrow YZ\} \models X \longrightarrow Y$ and $\{X \longrightarrow YZ\} \models X \longrightarrow Z$.

Assume $X \longrightarrow YZ$.

Apply *Reflexivity* on $Y \subseteq YZ$ to derive $YZ \longrightarrow Y$.
Apply *Transitivity* on $X \longrightarrow YZ$ and $YZ \longrightarrow Y$ to derive $X \longrightarrow Y$.

Hence, $X \longrightarrow Y$

The Decomposition rule

$$\text{if } X \longrightarrow YZ, \text{ then } X \longrightarrow Y \text{ and } X \longrightarrow Z.$$

We will show that the Decomposition rule is sound, but *not* independent.

How: Prove the Decomposition rule via Armstrong's Axioms

To prove: $\{X \longrightarrow YZ\} \models X \longrightarrow Y$ and $\{X \longrightarrow YZ\} \models X \longrightarrow Z$.

Assume $X \longrightarrow YZ$.

Apply *Reflexivity* on $Y \subseteq YZ$ to derive $YZ \longrightarrow Y$.
Apply *Transitivity* on $X \longrightarrow YZ$ and $YZ \longrightarrow Y$ to derive $X \longrightarrow Y$.

Hence, $X \longrightarrow Y$ ($X \longrightarrow Z$ is analogous).

The Decomposition rule

$$\text{if } X \longrightarrow YZ, \text{ then } X \longrightarrow Y \text{ and } X \longrightarrow Z.$$

We will show that the Decomposition rule is sound, but *not* independent.

How: Prove the Decomposition rule via Armstrong's Axioms

To prove: $\{X \longrightarrow YZ\} \models X \longrightarrow Y$ and $\{X \longrightarrow YZ\} \models X \longrightarrow Z$.

Assume $X \longrightarrow YZ$.

⋮

Apply *Reflexivity* on $Z \subseteq YZ$ to derive $YZ \longrightarrow Z$.
Apply *Transitivity* on $X \longrightarrow YZ$ and $YZ \longrightarrow Z$ to derive $X \longrightarrow Z$.

⋮

Hence, $X \longrightarrow Y$ ($X \longrightarrow Z$ is analogous).

Functional dependencies and the attribute closure

Consider a set of functional dependencies $\mathfrak{S}$ and a functional dependency $X \longrightarrow Y$.

- Can we automatically verify $\mathfrak{S} \models X \longrightarrow Y$?
- Can we automatically compute $\mathfrak{S} \models X \longrightarrow Y$?

Functional dependencies and the attribute closure

Consider a set of functional dependencies $\mathfrak{S}$ and a functional dependency $X \longrightarrow Y$.

- Can we automatically verify $\mathfrak{S} \models X \longrightarrow Y$?
- Can we automatically compute $\mathfrak{S} \models X \longrightarrow Y$?

$\textsc{Closure}(\mathfrak{S}, X)$

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

1: $\mathit{closure} := X$.
2: **while** there exists $(A \longrightarrow B) \in \mathfrak{S}$ with
  (I) $A \subseteq \mathit{closure}$; and
  (II) $B \not\subseteq \mathit{closure}$
 do
3:   $\mathit{closure} := \mathit{closure} \cup B$.
4: **return** $\mathit{closure}$.

Functional dependencies and the attribute closure

Consider a set of functional dependencies $\mathfrak{S}$ and a functional dependency $X \longrightarrow Y$.

- Can we automatically verify $\mathfrak{S} \models X \longrightarrow Y$?
- Can we automatically compute $\mathfrak{S} \models X \longrightarrow Y$?

$\textsc{Closure}(\mathfrak{S}, X)$

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

$\mathit{closure} := X$.
while there exists $(A \longrightarrow B) \in \mathfrak{S}$ with
  (I) $A \subseteq \mathit{closure}$; and
  (II) $B \not\subseteq \mathit{closure}$
do
  $\mathit{closure} := \mathit{closure} \cup B$.
return $\mathit{closure}$.

Correctness proof!

Correctness proof for CLOSURE

Correctness proof for Closure

Correctness proofs

- Does Closure terminate?
- Does Closure do what we want it to do?

Correctness proof for Closure

Correctness proofs

- Does Closure terminate?
- Does Closure do what we want it to do?

$$\text{If attribute } y \in \textsc{Closure}(\mathfrak{S}, X) \text{ then } \mathfrak{S} \models X \longrightarrow y.$$

Correctness proof for Closure

Correctness proofs

- Does Closure terminate?
- Does Closure do what we want it to do?

  If attribute $y \in \textsc{Closure}(\mathfrak{S}, X)$ then $\mathfrak{S} \models X \longrightarrow y$.
  If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Correctness proof for Closure

Correctness proofs

- Does Closure terminate?
- Does Closure do what we want it to do?

  Soundness If attribute $y \in \textsc{Closure}(\mathfrak{S}, X)$ then $\mathfrak{S} \models X \longrightarrow y$.

  Completeness If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Proof: Closure terminates

Closure($\mathfrak{S}$, X)

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

$closure := X$.
while there exists $(A \longrightarrow B) \in \mathfrak{S}$ with

(I) $A \subseteq closure$; and

(II) $B \not\subseteq closure$

do
  $closure := closure \cup B$.
return $closure$.

Proof: Closure terminates

Closure($\mathfrak{S}$, X)

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

$closure := X$.
while there exists $(A \longrightarrow B) \in \mathfrak{S}$ with $\longleftarrow$ *$\mathfrak{S}$ is a finite set.*
  (I) $A \subseteq closure$; and
  (II) $B \not\subseteq closure$
 do
  $closure := closure \cup B$.
return $closure$.

Proof: CLOSURE terminates

CLOSURE($\mathfrak{S}$, X)

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

$closure := X$.
while there exists $(A \longrightarrow B) \in \mathfrak{S}$ with ⟵ *$\mathfrak{S}$ is a finite set.*
  (I) $A \subseteq closure$; and
  (II) $B \nsubseteq closure$
 do
  $closure := closure \cup B$. ⟵ *Upper-bounded in size.*
return $closure$.

Proof: Closure is sound

"If attribute $y \in \textsc{Closure}(\mathfrak{S}, X)$ then $\mathfrak{S} \models X \longrightarrow y$."

$\textsc{Closure}(\mathfrak{S}, X)$

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

1: $\mathit{closure} := X$.

2: **while** there exists $(A \longrightarrow B) \in \mathfrak{S}$ with

(I) $A \subseteq \mathit{closure}$; and

(II) $B \not\subseteq \mathit{closure}$

do

3: $\mathit{closure} := \mathit{closure} \cup B$.

4: **return** $\mathit{closure}$.

Proof: Closure is sound

"If attribute $y \in \textsc{Closure}(\mathfrak{S}, X)$ then $\mathfrak{S} \models X \longrightarrow y$."

Closure($\mathfrak{S}$, X)

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

$\mathit{closure} := X$.
<u>Invariant</u>: If attribute $y \in \mathit{closure}$ then $\mathfrak{S} \models X \longrightarrow y$.
while there exists $(A \longrightarrow B) \in \mathfrak{S}$ with

(I) $A \subseteq \mathit{closure}$; and

(II) $B \not\subseteq \mathit{closure}$

do
$\quad \mathit{closure} := \mathit{closure} \cup B$.
return $\mathit{closure}$.

Proof: $\textsc{Closure}$ is sound

"If attribute $y \in \textsc{Closure}(\mathfrak{S}, X)$ then $\mathfrak{S} \models X \longrightarrow y$."

$\textsc{Closure}(\mathfrak{S}, X)$

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

$\mathit{closure} := X$.

<u>INVARIANT</u>: If attribute $y \in \mathit{closure}$ then $\mathfrak{S} \models X \longrightarrow y$.

while there exists $(A \longrightarrow B) \in \mathfrak{S}$ with

(I) $A \subseteq \mathit{closure}$; and

(II) $B \not\subseteq \mathit{closure}$

do

$\mathit{closure} := \mathit{closure} \cup B$.

return $\mathit{closure}$.

Proof: CLOSURE is sound

"If attribute $y \in \textsc{Closure}(\mathfrak{S}, X)$ then $\mathfrak{S} \models X \longrightarrow y$."

CLOSURE($\mathfrak{S}$, X)

Compute the set of all attributes y for which $\mathfrak{S} \models X \longrightarrow y$ holds.

1: $\mathit{closure} := X$. ✔

INVARIANT: If attribute $y \in \mathit{closure}$ then $\mathfrak{S} \models X \longrightarrow y$.

2: **while** there exists $(A \longrightarrow B) \in \mathfrak{S}$ with

(I) $A \subseteq \mathit{closure}$; and

(II) $B \not\subseteq \mathit{closure}$

do

3: $\mathit{closure} := \mathit{closure} \cup B$. ✔

4: **return** $\mathit{closure}$.

Proof: Closure is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$."

Proof: Closure is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$."

Proof by Contradiction

Proof: Closure is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \text{Closure}(\mathfrak{S}, X)$."

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \text{Closure}(\mathfrak{S}, X)$.

(proof details)

A contradiction. Hence, our original assumption was wrong, and we must have $y \in \text{Closure}(\mathfrak{S}, X)$.

Proof: Closure is complete

“If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$.”

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \textsc{Closure}(\mathfrak{S}, X)$.

Idea: construct an instance $\mathcal{I}$ that satisfies all of $\mathfrak{S}$, but not $X \longrightarrow y$.

A contradiction. Hence, our original assumption was wrong, and we must have $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Proof: Closure is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$."

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \textsc{Closure}(\mathfrak{S}, X)$.

Idea: construct an instance $\mathcal{I}$ that satisfies all of $\mathfrak{S}$, but not $X \longrightarrow y$.

Make $\mathcal{I}$ with rows r_1, r_2 such that $r_1[z] = r_2[z]$ if and only if $z \in \textsc{Closure}(\mathfrak{S}, X)$.

A contradiction. Hence, our original assumption was wrong,
and we must have $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Proof: Closure is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$."

$\in \textsc{Closure}(\mathfrak{S}, X)$				$\notin \textsc{Closure}(\mathfrak{S}, X)$			
c_1	c_2	$\dots$	c_n	o_1	o_2	$\dots$	o_m
1	1	$\dots$	1	0	0	$\dots$	0
1	1	$\dots$	1	1	1	$\dots$	1

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \textsc{Closure}(\mathfrak{S}, X)$.

Idea: construct an instance $\mathcal{I}$ that satisfies all of $\mathfrak{S}$, but not $X \longrightarrow y$.

Make $\mathcal{I}$ with rows r_1, r_2 such that $r_1[z] = r_2[z]$ if and only if $z \in \textsc{Closure}(\mathfrak{S}, X)$.

A contradiction. Hence, our original assumption was wrong, and we must have $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Proof: $\textsc{Closure}$ is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$."

$\in \textsc{Closure}(\mathfrak{S}, X)$				$\notin \textsc{Closure}(\mathfrak{S}, X)$			
c_1	c_2	$\dots$	c_n	o_1	o_2	$\dots$	o_m
1	1	$\dots$	1	0	0	$\dots$	0
1	1	$\dots$	1	1	1	$\dots$	1

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \textsc{Closure}(\mathfrak{S}, X)$.

Idea: construct an instance $\mathcal{I}$ that satisfies all of $\mathfrak{S}$, but not $X \longrightarrow y$.

Make $\mathcal{I}$ with rows r_1, r_2 such that $r_1[z] = r_2[z]$ if and only if $z \in \textsc{Closure}(\mathfrak{S}, X)$.

Let $A \longrightarrow B \in \mathfrak{S}$. Two cases: $A \subseteq \textsc{Closure}(\mathfrak{S}, X)$ and $A \not\subseteq \textsc{Closure}(\mathfrak{S}, X)$.

A contradiction. Hence, our original assumption was wrong, and we must have $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Proof: Closure is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$."

$\in \textsc{Closure}(\mathfrak{S}, X)$				$\notin \textsc{Closure}(\mathfrak{S}, X)$			
c_1	c_2	$\dots$	c_n	o_1	o_2	$\dots$	o_m
1	1	$\dots$	1	0	0	$\dots$	0
1	1	$\dots$	1	1	1	$\dots$	1

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \textsc{Closure}(\mathfrak{S}, X)$.

Idea: construct an instance $\mathcal{I}$ that satisfies all of $\mathfrak{S}$, but not $X \longrightarrow y$.

Make $\mathcal{I}$ with rows r_1, r_2 such that $r_1[z] = r_2[z]$ if and only if $z \in \textsc{Closure}(\mathfrak{S}, X)$.

Let $A \longrightarrow B \in \mathfrak{S}$. Two cases: $A \subseteq \textsc{Closure}(\mathfrak{S}, X)$ and $A \nsubseteq \textsc{Closure}(\mathfrak{S}, X)$.

$A \subseteq \textsc{Closure}(\mathfrak{S}, X)$: Line 3 of Closure ($\mathit{closure} := \mathit{closure} \cup B$).

A contradiction. Hence, our original assumption was wrong, and we must have $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Proof: Closure is complete

“If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$.”

$\in \textsc{Closure}(\mathfrak{S}, X)$				$\notin \textsc{Closure}(\mathfrak{S}, X)$			
c_1	c_2	$\dots$	c_n	o_1	o_2	$\dots$	o_m
1	1	$\dots$	1	0	0	$\dots$	0
1	1	$\dots$	1	1	1	$\dots$	1

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \textsc{Closure}(\mathfrak{S}, X)$.

Idea: construct an instance $\mathcal{I}$ that satisfies all of $\mathfrak{S}$, but not $X \longrightarrow y$.

Make $\mathcal{I}$ with rows r_1, r_2 such that $r_1[z] = r_2[z]$ if and only if $z \in \textsc{Closure}(\mathfrak{S}, X)$.

Let $A \longrightarrow B \in \mathfrak{S}$. Two cases: $A \subseteq \textsc{Closure}(\mathfrak{S}, X)$ and $A \not\subseteq \textsc{Closure}(\mathfrak{S}, X)$.

$A \subseteq \textsc{Closure}(\mathfrak{S}, X)$: Line 3 of Closure (*closure* := *closure* $\cup B$).

$A \not\subseteq \textsc{Closure}(\mathfrak{S}, X)$: Automatically holds.

A contradiction. Hence, our original assumption was wrong, and we must have $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Proof: $\textsc{Closure}$ is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$."

$\in \textsc{Closure}(\mathfrak{S}, X)$				$\notin \textsc{Closure}(\mathfrak{S}, X)$			
c_1	c_2	$\dots$	c_n	o_1	o_2	$\dots$	o_m
1	1	$\dots$	1	0	0	$\dots$	0
1	1	$\dots$	1	1	1	$\dots$	1

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \textsc{Closure}(\mathfrak{S}, X)$.

Idea: construct an instance $\mathcal{I}$ that satisfies all of $\mathfrak{S}$, but not $X \longrightarrow y$.

Make $\mathcal{I}$ with rows r_1, r_2 such that $r_1[z] = r_2[z]$ if and only if $z \in \textsc{Closure}(\mathfrak{S}, X)$.

Let $A \longrightarrow B \in \mathfrak{S}$. Two cases: $A \subseteq \textsc{Closure}(\mathfrak{S}, X)$ and $A \nsubseteq \textsc{Closure}(\mathfrak{S}, X)$.

$A \subseteq \textsc{Closure}(\mathfrak{S}, X)$: Line 3 of $\textsc{Closure}$ ($\mathit{closure} := \mathit{closure} \cup B$).

$A \nsubseteq \textsc{Closure}(\mathfrak{S}, X)$: Automatically holds.

$X \subseteq \textsc{Closure}(\mathfrak{S}, X)$, $y \notin \textsc{Closure}(\mathfrak{S}, X)$. Hence, $\mathcal{I}$ does not satisfy $X \longrightarrow y$.

A contradiction. Hence, our original assumption was wrong, and we must have $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Proof: Closure is complete

"If y is an attribute and $\mathfrak{S} \models X \longrightarrow y$, then $y \in \textsc{Closure}(\mathfrak{S}, X)$."

$\in \textsc{Closure}(\mathfrak{S}, X)$				$\notin \textsc{Closure}(\mathfrak{S}, X)$			
c_1	c_2	$\dots$	c_n	o_1	o_2	$\dots$	o_m
1	1	$\dots$	1	0	0	$\dots$	0
1	1	$\dots$	1	1	1	$\dots$	1

Proof by Contradiction

Assume $\mathfrak{S} \models X \longrightarrow y$ and $y \notin \textsc{Closure}(\mathfrak{S}, X)$.

Idea: construct an instance $\mathcal{I}$ that satisfies all of $\mathfrak{S}$, but not $X \longrightarrow y$.

Make $\mathcal{I}$ with rows r_1, r_2 such that $r_1[z] = r_2[z]$ if and only if $z \in \textsc{Closure}(\mathfrak{S}, X)$.

Let $A \longrightarrow B \in \mathfrak{S}$. Two cases: $A \subseteq \textsc{Closure}(\mathfrak{S}, X)$ and $A \nsubseteq \textsc{Closure}(\mathfrak{S}, X)$.

$A \subseteq \textsc{Closure}(\mathfrak{S}, X)$: Line 3 of Closure ($\mathit{closure} := \mathit{closure} \cup B$).

$A \nsubseteq \textsc{Closure}(\mathfrak{S}, X)$: Automatically holds.

$X \subseteq \textsc{Closure}(\mathfrak{S}, X)$, $y \notin \textsc{Closure}(\mathfrak{S}, X)$. Hence, $\mathcal{I}$ does not satisfy $X \longrightarrow y$.

We conclude that $\mathfrak{S} \models X \longrightarrow y$ *cannot* hold.

A contradiction. Hence, our original assumption was wrong, and we must have $y \in \textsc{Closure}(\mathfrak{S}, X)$.

Functional dependencies and the attribute closure—results

Consider a set of functional dependencies $\mathfrak{S}$ and a functional dependency $X \longrightarrow Y$.

- Can we automatically verify $\mathfrak{S} \models X \longrightarrow Y$?
- Can we automatically compute $\mathfrak{S} \models X \longrightarrow Y$?

Functional dependencies and the attribute closure—results

Consider a set of functional dependencies $\mathfrak{S}$ and a functional dependency $X \longrightarrow Y$.

- Can we automatically verify $\mathfrak{S} \models X \longrightarrow Y$?
  Yes. Verify whether $Y \subseteq \textsc{Closure}(\mathfrak{S}, X)$.

- Can we automatically compute $\mathfrak{S} \models X \longrightarrow Y$?

Functional dependencies and the attribute closure—results

Consider a set of functional dependencies $\mathfrak{S}$ and a functional dependency $X \longrightarrow Y$.

- Can we automatically verify $\mathfrak{S} \models X \longrightarrow Y$?
  Yes. Verify whether $Y \subseteq \textsc{Closure}(\mathfrak{S}, X)$.

- Can we automatically compute $\mathfrak{S} \models X \longrightarrow Y$?
  Yes. For every X, compute $\textsc{Closure}(\mathfrak{S}, X)$.

Functional dependencies and the attribute closure—results

Consider a set of functional dependencies $\mathfrak{S}$ and a functional dependency $X \longrightarrow Y$.

- Can we automatically verify $\mathfrak{S} \models X \longrightarrow Y$?
  Yes. Verify whether $Y \subseteq \textsc{Closure}(\mathfrak{S}, X)$.

- Can we automatically compute $\mathfrak{S} \models X \longrightarrow Y$?
  Yes. For every X, compute $\textsc{Closure}(\mathfrak{S}, X)$.

We typically write X^+ to denote $\textsc{Closure}(\mathfrak{S}, X)$ if $\mathfrak{S}$ is clear from the context.

Closure of functional dependencies

Definition

Let $\mathfrak{S}$ be a set of functional dependencies. The *closure* of $\mathfrak{S}$ is

$$\mathfrak{S}^+ = \{X \longrightarrow Y \mid \mathfrak{S} \models X \longrightarrow Y\}.$$

Closure of functional dependencies

Definition

Let $\mathfrak{S}$ be a set of functional dependencies. The *closure* of $\mathfrak{S}$ is

$$\mathfrak{S}^+ = \{X \longrightarrow Y \mid \mathfrak{S} \models X \longrightarrow Y\}.$$

Example: **student**(sid, name, age, birthdate, program, department)

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Compute $\mathfrak{S}^+$ with $\mathfrak{S} = \{S \longrightarrow NABPD, B \longrightarrow A, P \longrightarrow D\}$.

Closure of functional dependencies

Definition

Let $\mathfrak{S}$ be a set of functional dependencies. The *closure* of $\mathfrak{S}$ is

$$\mathfrak{S}^+ = \{X \longrightarrow Y \mid \mathfrak{S} \models X \longrightarrow Y\}.$$

Example: **student**(<u>sid</u>, name, age, birthdate, program, department)

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Compute $\mathfrak{S}^+$ with $\mathfrak{S} = \{S \longrightarrow NABPD, B \longrightarrow A, P \longrightarrow D\}$.
How: Compute $\textsc{Closure}(\mathfrak{S}, X)$ for every $X \subseteq \{S, N, A, B, P, D\}$.

Closure of functional dependencies

Definition

Let $\mathfrak{S}$ be a set of functional dependencies. The *closure* of $\mathfrak{S}$ is

$$\mathfrak{S}^+ = \{X \longrightarrow Y \mid \mathfrak{S} \models X \longrightarrow Y\}.$$

Example: **student**(sid, name, age, birthdate, program, department)

I will use shorthand notations A (age), P (program), B (birthdate), and D (department).

Compute $\mathfrak{S}^+$ with $\mathfrak{S} = \{S \longrightarrow NABPD, B \longrightarrow A, P \longrightarrow D\}$.

- We have $X \longrightarrow Y$ for all $Y \subseteq X \subseteq \{S, N, A, B, P, D\}$.
- We have $SX \longrightarrow Y$ for all $X, Y \subseteq \{S, N, A, B, P, D\}$.
- We have $BX \longrightarrow AY$ for all $Y \subseteq X \subseteq \{S, N, A, B, P, D\}$.
- We have $PX \longrightarrow DY$ for all $Y \subseteq X \subseteq \{S, N, A, B, P, D\}$.
- We have $BPX \longrightarrow ADY$ for all $Y \subseteq X \subseteq \{S, N, A, B, P, D\}$.

Minimal Cover for functional dependencies

Definition

Let $\mathfrak{S}$ be a set of functional dependencies.

A *minimal cover* of $\mathfrak{S}$ is a set S of functional dependencies such that:

Minimal Cover for functional dependencies

Definition

Let $\mathfrak{S}$ be a set of functional dependencies.
A *minimal cover* of $\mathfrak{S}$ is a set S of functional dependencies such that:

1. $\mathfrak{S}^+ = S^+$.
   ($\mathfrak{S}$ and S describe the *same* functional dependencies).

Minimal Cover for functional dependencies

Definition

Let $\mathfrak{S}$ be a set of functional dependencies.

A *minimal cover* of $\mathfrak{S}$ is a set S of functional dependencies such that:

1. $\mathfrak{S}^+ = S^+$.

   ($\mathfrak{S}$ and S describe the *same* functional dependencies).

2. All functional dependencies $X \longrightarrow Y \in S$ must have $|Y| = 1$.

   (Functional dependencies in S are *minimalistic*).

Minimal Cover for functional dependencies

Definition

Let $\mathfrak{S}$ be a set of functional dependencies.

A *minimal cover* of $\mathfrak{S}$ is a set S of functional dependencies such that:

1. $\mathfrak{S}^+ = S^+$.

   ($\mathfrak{S}$ and S describe the *same* functional dependencies).
2. All functional dependencies $X \longrightarrow Y \in S$ must have $|Y| = 1$.

   (Functional dependencies in S are *minimalistic*).
3. For every $R \subset S$, we have $R^+ \neq S^+$.

   (All of S is necessary to describe the *same* functional dependencies as $\mathfrak{S}$).

Beyond functional dependencies

- Multivalued dependencies.
- Join dependencies.
- Inclusion dependencies.
- ....

Multivalued dependency over relation schema **R**

$$X \twoheadrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{; } Z \text{ the remaining attributes of } \mathbf{R}\text{).}$$

Informal

"Given a value for the X attributes, the values for attributes Y and Z are independent."

Multivalued dependency over relation schema **R**

$$X \twoheadrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}; Z \text{ the remaining attributes of } \mathbf{R}\text{).}$$

Informal

"Given a value for the X attributes, the values for attributes Y and Z are independent."
If two rows in an instance of **R** have the same values for attributes X,
Then there must be a third row with the same values for attributes X, and

- the values for attributes Y from the first row, and
- the values for attributes Z from the second row.

Multivalued dependency over relation schema **R**

$$X \twoheadrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{; } Z \text{ the remaining attributes of } \mathbf{R}\text{).}$$

Informal

"Given a value for the X attributes, the values for attributes Y and Z are independent."
If two rows in an instance of **R** have the same values for attributes X,
Then there must be a third row with the same values for attributes X, and

- the values for attributes Y from the first row, and
- the values for attributes Z from the second row.

Formal

For every instance $\mathcal{I}$ of **R** and every pair of rows $r_1, r_2 \in \mathcal{I}$ with $r_1[X] = r_2[X]$,
there exists a row $r_3 \in \mathcal{I}$ with $r_1[XY] = r_3[XY]$ and $r_2[XZ] = r_3[XZ]$.

Multivalued dependency over relation schema **R**

$X \twoheadrightarrow Y$ (with X and Y attributes of **R**; Z the remaining attributes of **R**).

Formal

For every instance $\mathcal{I}$ of **R** and every pair of rows $r_1, r_2 \in \mathcal{I}$ with $r_1[X] = r_2[X]$, there exists a row $r_3 \in \mathcal{I}$ with $r_1[XY] = r_3[XY]$ and $r_2[XZ] = r_3[XZ]$.

x_1	x_2	y_1	y_2	z_1	z_2
1	2	A	B	α	β
1	2	C	D	γ	δ

Multivalued dependency over relation schema **R**

$X \twoheadrightarrow Y$ (with X and Y attributes of **R**; Z the remaining attributes of **R**).

Formal

For every instance $\mathcal{I}$ of **R** and every pair of rows $r_1, r_2 \in \mathcal{I}$ with $r_1[X] = r_2[X]$, there exists a row $r_3 \in \mathcal{I}$ with $r_1[XY] = r_3[XY]$ and $r_2[XZ] = r_3[XZ]$.

	x_1	x_2	y_1	y_2	z_1	z_2
$r_1 \longrightarrow$	1	2	A	B	α	β
$r_2 \longrightarrow$	1	2	C	D	γ	δ
	1	2	A	B	γ	δ

Multivalued dependency over relation schema **R**

$X \twoheadrightarrow Y$ (with X and Y attributes of $\mathbf{R}$; Z the remaining attributes of $\mathbf{R}$).

Formal

For every instance $\mathcal{I}$ of $\mathbf{R}$ and every pair of rows $r_1, r_2 \in \mathcal{I}$ with $r_1[X] = r_2[X]$, there exists a row $r_3 \in \mathcal{I}$ with $r_1[XY] = r_3[XY]$ and $r_2[XZ] = r_3[XZ]$.

	x_1	x_2	y_1	y_2	z_1	z_2
$r_2 \longrightarrow$	1	2	A	B	α	β
$r_1 \longrightarrow$	1	2	C	D	γ	δ
	1	2	A	B	γ	δ
	1	2	C	D	α	β

Multivalued dependency over relation schema **R**

$$X \twoheadrightarrow Y \qquad \text{(with } X \text{ and } Y \text{ attributes of } \mathbf{R}\text{; } Z \text{ the remaining attributes of } \mathbf{R}\text{).}$$

Formal

For every instance $\mathcal{I}$ of **R** and every pair of rows $r_1, r_2 \in \mathcal{I}$ with $r_1[X] = r_2[X]$, there exists a row $r_3 \in \mathcal{I}$ with $r_1[XY] = r_3[XY]$ and $r_2[XZ] = r_3[XZ]$.

x_1	x_2	y_1	y_2	z_1	z_2
1	2	A	B	α	β
1	2	C	D	γ	δ
1	2	A	B	γ	δ
1	2	C	D	α	β

=

x_1	x_2	y_1	y_2
1	2	A	B
1	2	C	D

$\bowtie$

x_1	x_2	z_1	z_2
1	2	α	β
1	2	γ	δ

An example of multivalued dependencies

course	**student**	**TA**
Programming	Celeste	Alicia
Programming	Frieda	Alicia
Programming	Celeste	Dafni
Programming	Frieda	Dafni
Databases	Bo	Eva
Databases	Dafni	Eva
Databases	Bo	Alicia
Databases	Dafni	Alicia

An example of multivalued dependencies

course	**student**	**TA**
Programming	Celeste	Alicia
Programming	Frieda	Alicia
Programming	Celeste	Dafni
Programming	Frieda	Dafni
Databases	Bo	Eva
Databases	Dafni	Eva
Databases	Bo	Alicia
Databases	Dafni	Alicia

The enrolled students of a course are *independent* of the TAs.

An example of multivalued dependencies

course	**student**	**TA**
Programming	Celeste	Alicia
Programming	Frieda	Alicia
Programming	Celeste	Dafni
Programming	Frieda	Dafni
Databases	Bo	Eva
Databases	Dafni	Eva
Databases	Bo	Alicia
Databases	Dafni	Alicia

The enrolled students of a course are *independent* of the TAs.

"course $\twoheadrightarrow$ student" and "course $\twoheadrightarrow$ TA".

Reasoning with functional and multivalued dependencies

Reasoning with functional and multivalued dependencies

Armstrong's Axioms.

Reasoning with functional and multivalued dependencies

Armstrong's Axioms.

Complementation If $X \twoheadrightarrow Y$, then $X \twoheadrightarrow Z$ (with Z all attributes of $\mathbf{R}$ not in X and Y).

Reasoning with functional and multivalued dependencies

Armstrong's Axioms.

Complementation	If $X \twoheadrightarrow Y$, then $X \twoheadrightarrow Z$ (with Z all attributes of $\mathbf{R}$ not in X and Y).
Augmentation	If $X \twoheadrightarrow Y$ and $V \subseteq W$, then $XW \twoheadrightarrow YV$.

Reasoning with functional and multivalued dependencies

Armstrong's Axioms.

Complementation	If $X \twoheadrightarrow Y$, then $X \twoheadrightarrow Z$ (with Z all attributes of $\mathbf{R}$ not in X and Y).
Augmentation	If $X \twoheadrightarrow Y$ and $V \subseteq W$, then $XW \twoheadrightarrow YV$.
Transitivity	If $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$, then $X \twoheadrightarrow (Z \setminus Y)$.

Reasoning with functional and multivalued dependencies

Armstrong's Axioms.

Complementation If $X \twoheadrightarrow Y$, then $X \twoheadrightarrow Z$ (with Z all attributes of **R** not in X and Y).

Augmentation If $X \twoheadrightarrow Y$ and $V \subseteq W$, then $XW \twoheadrightarrow YV$.

Transitivity If $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$, then $X \twoheadrightarrow (Z \setminus Y)$.

Replication If $X \longrightarrow Y$, then $X \twoheadrightarrow Y$.

Reasoning with functional and multivalued dependencies

Armstrong's Axioms.

Complementation	If $X \twoheadrightarrow Y$, then $X \twoheadrightarrow Z$ (with Z all attributes of $\mathbf{R}$ not in X and Y).
Augmentation	If $X \twoheadrightarrow Y$ and $V \subseteq W$, then $XW \twoheadrightarrow YV$.
Transitivity	If $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$, then $X \twoheadrightarrow (Z \setminus Y)$.
Replication	If $X \longrightarrow Y$, then $X \twoheadrightarrow Y$.
Coalescence	If $X \twoheadrightarrow Y$ and $V \longrightarrow W$ such that $Y \cap V = \emptyset$ and $W \subseteq Y$, then $X \longrightarrow W$.

Soundness of the Coalescence rule

$$\text{If } X \twoheadrightarrow Y \text{ and } V \longrightarrow W \text{ such that } Y \cap V = \emptyset \text{ and } W \subseteq Y \text{, then } X \longrightarrow W.$$

Proof

Let **R** be a relational schema that satisfies the premise of the Coalescence rule.
By definition: we have $X \longrightarrow W$ if we have $r_1[X] = r_2[X] \implies r_1[W] = r_2[W]$
for every pair of rows r_1, r_2 in every instance $\mathcal{I}$ of **R**.

Soundness of the Coalescence rule

$$\text{If } X \twoheadrightarrow Y \text{ and } V \longrightarrow W \text{ such that } Y \cap V = \emptyset \text{ and } W \subseteq Y, \text{ then } X \longrightarrow W.$$

Proof

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of an instance $\mathcal{I}$ of a relational schema that satisfies the premise of the Coalescence rule such that $r_1[X] = r_2[X]$.

Soundness of the Coalescence rule

If $X \twoheadrightarrow Y$ and $V \longrightarrow W$ such that $Y \cap V = \emptyset$ and $W \subseteq Y$, then $X \longrightarrow W$.

Proof

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of an instance $\mathcal{I}$ of a relational schema that satisfies the premise of the Coalescence rule such that $r_1[X] = r_2[X]$.

(proof details)

Hence, $r_1[W] = r_2[W]$ holds.

Soundness of the Coalescence rule

If $X \twoheadrightarrow Y$ and $V \longrightarrow W$ such that $Y \cap V = \emptyset$ and $W \subseteq Y$, then $X \longrightarrow W$.

Proof

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of an instance $\mathcal{I}$ of a relational schema that satisfies the premise of the Coalescence rule such that $r_1[X] = r_2[X]$.

By $X \twoheadrightarrow Y$ and $r_1[X] = r_2[X]$, there exists a row $r_3 \in \mathcal{I}$ such that
$r_1[XY] = r_3[XY]$ and $r_2[XZ] = r_3[XZ]$.

Hence, $r_1[W] = r_2[W]$ holds.

Soundness of the Coalescence rule

$$\text{If } X \twoheadrightarrow Y \text{ and } V \longrightarrow W \text{ such that } Y \cap V = \emptyset \text{ and } W \subseteq Y\text{, then } X \longrightarrow W.$$

Proof

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of an instance $\mathcal{I}$ of a relational schema that satisfies the premise of the Coalescence rule such that $r_1[X] = r_2[X]$.

By $X \twoheadrightarrow Y$ and $r_1[X] = r_2[X]$, there exists a row $r_3 \in \mathcal{I}$ such that

$$r_1[XY] = r_3[XY] \text{ and } r_2[XZ] = r_3[XZ].$$

As $Y \cap V = \emptyset$, we have $V \subseteq XZ$ (with Z all attributes of **R** not in X and Y).

Hence, $r_1[W] = r_2[W]$ holds.

Soundness of the Coalescence rule

If $X \twoheadrightarrow Y$ and $V \longrightarrow W$ such that $Y \cap V = \emptyset$ and $W \subseteq Y$, then $X \longrightarrow W$.

Proof

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of an instance $\mathcal{I}$ of a relational schema that satisfies the premise of the Coalescence rule such that $r_1[X] = r_2[X]$.

By $X \twoheadrightarrow Y$ and $r_1[X] = r_2[X]$, there exists a row $r_3 \in \mathcal{I}$ such that

$$r_1[XY] = r_3[XY] \text{ and } r_2[XZ] = r_3[XZ].$$

As $Y \cap V = \emptyset$, we have $V \subseteq XZ$ (with Z all attributes of $\mathbf{R}$ not in X and Y).

As $r_2[XZ] = r_3[XZ]$, $V \subseteq XZ$, and $V \longrightarrow W$, we have $r_2[W] = r_3[W]$.

Hence, $r_1[W] = r_2[W]$ holds.

Soundness of the Coalescence rule

If $X \twoheadrightarrow Y$ and $V \longrightarrow W$ such that $Y \cap V = \emptyset$ and $W \subseteq Y$, then $X \longrightarrow W$.

Proof

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of an instance $\mathcal{I}$ of a relational schema that satisfies the premise of the Coalescence rule such that $r_1[X] = r_2[X]$.

By $X \twoheadrightarrow Y$ and $r_1[X] = r_2[X]$, there exists a row $r_3 \in \mathcal{I}$ such that

$$r_1[XY] = r_3[XY] \text{ and } r_2[XZ] = r_3[XZ].$$

As $Y \cap V = \emptyset$, we have $V \subseteq XZ$ (with Z all attributes of $\mathbf{R}$ not in X and Y).
As $r_2[XZ] = r_3[XZ]$, $V \subseteq XZ$, and $V \longrightarrow W$, we have $r_2[W] = r_3[W]$.
As $W \subseteq Y$ and $r_3[XY] = r_1[XY]$, we have $r_3[W] = r_1[W]$.

Hence, $r_1[W] = r_2[W]$ holds.

Join dependencies over relational schema $\mathbf{R}$

$$\bowtie\{X_1, X_2, \ldots, X_n\} \qquad \text{(with } X_i,\ 1 \leq i \leq n\text{, attributes of } \mathbf{R}\text{)}$$

Informal

"The projections $\pi_{X_i}(\mathbf{R})$, $1 \leq i \leq n$, are independent of each other."

Join dependencies over relational schema $\mathbf{R}$

$$\bowtie\{X_1, X_2, \ldots, X_n\} \qquad \text{(with } X_i,\ 1 \le i \le n\text{, attributes of } \mathbf{R}\text{)}$$

Informal

"The projections $\pi_{X_i}(\mathbf{R})$, $1 \le i \le n$, are independent of each other."

For any instance of $\mathbf{R}$, we obtain exactly that instance if we:

- first break-up the instance into its projections on X_i, $1 \le i \le n$, and
- then recombine these projections using the natural join.

Join dependencies over relational schema $\mathbf{R}$

$$\bowtie\{X_1, X_2, \dots, X_n\} \qquad \text{(with } X_i,\ 1 \le i \le n\text{, attributes of } \mathbf{R}\text{)}$$

Informal

"The projections $\pi_{X_i}(\mathbf{R})$, $1 \le i \le n$, are independent of each other."

For any instance of $\mathbf{R}$, we obtain exactly that instance if we:

- first break-up the instance into its projections on X_i, $1 \le i \le n$, and
- then recombine these projections using the natural join.

Generalization of multivalued dependencies:

$X \twoheadrightarrow Y$ is equivalent to $\bowtie\{XY, XZ\}$ (with Z all attributes of $\mathbf{R}$ not in X and Y).

Join dependencies over relational schema $\mathbf{R}$

$$\bowtie\{X_1, X_2, \dots, X_n\} \qquad \text{(with } X_i,\ 1 \leq i \leq n\text{, attributes of } \mathbf{R}\text{)}$$

Informal

"The projections $\pi_{X_i}(\mathbf{R})$, $1 \leq i \leq n$, are independent of each other."

For any instance of $\mathbf{R}$, we obtain exactly that instance if we:

- first break-up the instance into its projections on X_i, $1 \leq i \leq n$, and
- then recombine these projections using the natural join.

Generalization of multivalued dependencies:

$X \twoheadrightarrow Y$ is equivalent to $\bowtie\{XY, XZ\}$ (with Z all attributes of $\mathbf{R}$ not in X and Y).

Formal

We must have

$$\pi_{X_1}(\mathbf{R}) \bowtie \pi_{X_2}(\mathbf{R}) \bowtie \dots \bowtie \pi_{X_n}(\mathbf{R}) = \mathbf{R}.$$

An example of join dependencies

course (C)	**student** (S)	**TA** (T)	**Instructor** (I)
Databases	Bo	Eva	Celeste
Databases	Dafni	Eva	Celeste
Databases	Bo	Alicia	Celeste
Databases	Dafni	Alicia	Celeste
Databases	Bo	Eva	Frieda
Databases	Dafni	Eva	Frieda
Databases	Bo	Alicia	Frieda
Databases	Dafni	Alicia	Frieda

An example of join dependencies

course (C)	**student** (S)	**TA** (T)	**Instructor** (I)
Databases	Bo	Eva	Celeste
Databases	Dafni	Eva	Celeste
Databases	Bo	Alicia	Celeste
Databases	Dafni	Alicia	Celeste
Databases	Bo	Eva	Frieda
Databases	Dafni	Eva	Frieda
Databases	Bo	Alicia	Frieda
Databases	Dafni	Alicia	Frieda

The enrolled students, the TAs, and the Instructors are *independent*.

An example of join dependencies

course (C)	**student** (S)	**TA** (T)	**Instructor** (I)
Databases	Bo	Eva	Celeste
Databases	Dafni	Eva	Celeste
Databases	Bo	Alicia	Celeste
Databases	Dafni	Alicia	Celeste
Databases	Bo	Eva	Frieda
Databases	Dafni	Eva	Frieda
Databases	Bo	Alicia	Frieda
Databases	Dafni	Alicia	Frieda

The enrolled students, the TAs, and the Instructors are *independent*.

$$\bowtie\{CS, CT, CI\}.$$

Inclusion dependencies over relational schemas $\mathbf{R}_1$ and $\mathbf{R}_2$

$$\mathbf{R}_1[X] \subseteq \mathbf{R}_2[Y] \text{ (with } X \text{ attributes of } \mathbf{R}_1,\ Y \text{ attributes of } \mathbf{R}_2\text{)}$$

Informal

"Values for attributes X in $\mathbf{R}_1$ must also occur as values for attributes Y in $\mathbf{R}_2$".

Inclusion dependencies over relational schemas $\mathbf{R}_1$ and $\mathbf{R}_2$

$$\mathbf{R}_1[X] \subseteq \mathbf{R}_2[Y] \text{ (with } X \text{ attributes of } \mathbf{R}_1, Y \text{ attributes of } \mathbf{R}_2\text{)}$$

Informal

"Values for attributes X in $\mathbf{R}_1$ must also occur as values for attributes Y in $\mathbf{R}_2$".
We must have

$$\pi_X(\mathbf{R}_1) \subseteq \pi_Y(\mathbf{R}_2).$$

Inclusion dependencies over relational schemas $\mathbf{R}_1$ and $\mathbf{R}_2$

$$\mathbf{R}_1[X] \subseteq \mathbf{R}_2[Y] \text{ (with } X \text{ attributes of } \mathbf{R}_1, Y \text{ attributes of } \mathbf{R}_2)$$

Informal

"Values for attributes X in $\mathbf{R}_1$ must also occur as values for attributes Y in $\mathbf{R}_2$".
We must have

$$\pi_X(\mathbf{R}_1) \subseteq \pi_Y(\mathbf{R}_2).$$

Generalization of foreign key constraints.

Inclusion dependencies over relational schemas $\mathbf{R}_1$ and $\mathbf{R}_2$

$$\mathbf{R}_1[X] \subseteq \mathbf{R}_2[Y] \text{ (with } X \text{ attributes of } \mathbf{R}_1,\ Y \text{ attributes of } \mathbf{R}_2\text{)}$$

Informal

"Values for attributes X in $\mathbf{R}_1$ must also occur as values for attributes Y in $\mathbf{R}_2$".
We must have

$$\pi_X(\mathbf{R}_1) \subseteq \pi_Y(\mathbf{R}_2).$$

Generalization of foreign key constraints.

Formal

For every instance $\mathcal{I}_1$ of $\mathbf{R}_1$ and every row $r_1 \in \mathcal{I}$,
there exists a row in instance $\mathcal{I}_2$ of $\mathbf{R}_2$ with $r_1[X] = r_2[Y]$.

An example of inclusion dependencies

courses

cid	title	lecturer
2	Discrete Mathematics	3
3	Databases	2

faculty

fid	name	rank
2	Bo	Assistant
3	Celeste	Associate

An example of inclusion dependencies

courses		
cid	title	lecturer
2	Discrete Mathematics	3
3	Databases	2

faculty		
fid	name	rank
2	Bo	Assistant
3	Celeste	Associate

$$\textbf{courses}[\text{lecturer}] \subseteq \textbf{faculty}[\text{fid}].$$

Next: Decomposition and normal forms

How can we use dependency theory to:

- validate the quality of relational schemas,
- improve the quality of relational schemas.